

Welcome to Software Foundations CSE 5910

`www.eecs.yorku.ca/course/5910`

- **Name:** Franck van Breugel
- **Email:** franck@cse.yorku.ca
- **Office:** Lassonde Building, room 3046
- **Office hours:** Mondays, 16:00-17:00, Wednesdays, 16:00-17:00
or by appointment

Tests and a project (details on Wednesday).

Practice, practice, practice, ...

- weekly programming exercises
- programming exercises in the textbook

Drop deadline

November 7

Until this date you can drop the course without getting a grade for it and, hence, it will not affect your gpa.

www.registrar.yorku.ca/enrol/dates/fw14.htm contains important dates.

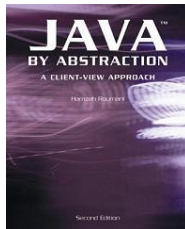
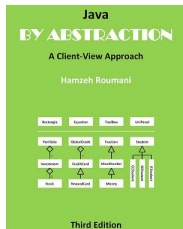
“If you put your name on something, then it is your work, unless you explicitly say that it is not.”

<http://gradstudies.yorku.ca/current-students/regulations/academic-honesty/> contains more details.

Hamzeh Roumani. *Java by Abstraction: A Client-View Approach*. Second, third or fourth edition. Pearson, Toronto. 2007, 2010 and 2014.

Why this textbook?

- As far as I know, this is the only textbook that uses the client-view (more about this view later in the course).
- The author is an award winning lecturer and teaches at York.



Each chapter consists of

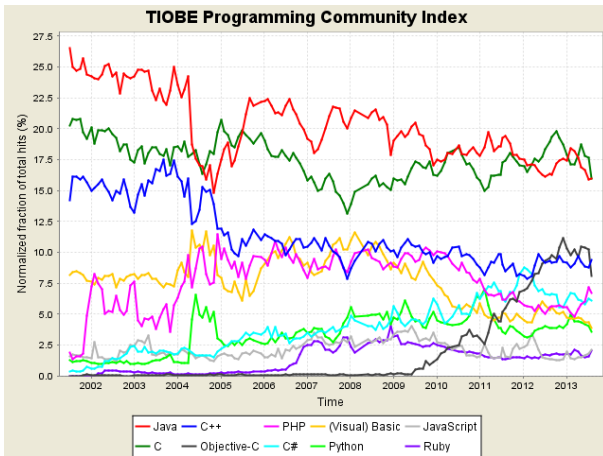
- reading material and contains several types of “boxes”
 - Programming Tip: essential
 - Java Details: useful
 - In More Depth: read and try to understand (don't worry if you don't understand them completely)
- review questions (try some yourself after having read a chapter),
- a lab (do it yourself),
- exercises (try some yourself, good preparation for tests), and
- eChecks.

Programming languages

ABC, Ada, Algol 60, Algol 68, Alice, APL, Basic, BPEL, C, C++, C#, Caml, COBOL, CSP, Eiffel, Emacs Lisp, Erlang, Esterel, Fortran, Haskell, IMP, Java, JavaScript, LaTeX, Lisp, LOTUS, Lustre, Maple, Mathematica, MATLAB, Mesa, Metafont, Miranda, ML, Modula-2, Oak, Oberon, Objective Caml, occam, Pascal, Perl, PHP, Pict, Pizza, PL/I, PostScript, Prolog, Promela, Python, Scala, Scheme, Simula, Smalltalk, SNOBOL, SOAP, Tcl, TeX, Turing, Visual Basic, Z, ...

Why Java?

Java is well-designed and popular.



source: www.tiobe.com

Why Java?

Because Java is popular, there are

- many textbooks about Java,
- many web pages about Java, and
- many software packages written in Java.

Also, one can easily invoke Matlab code within a Java program. Furthermore, since the Matlab interpreter is written in Java, one can directly invoke Java code in Matlab.

... this is *not* a course about Java.

Java is used to introduce you to object oriented programming and the basic principles of software development.

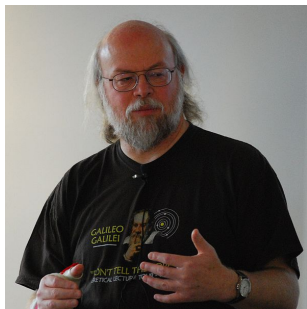
In the early 1990s, James Gosling and some of his colleagues at Sun Microsystems, developed a programming language to program device controllers. The language was called Oak after an oak tree that stood outside Gosling's office. The language was expanded to a general purpose programming language and was renamed Java (because Oak was already a trademark). On May 23, 1995, Java was announced.

“Even though the Web had been around for 20 years or so, with FTP and telnet, it was difficult to use. Then Mosaic came out in 1993 as an easy-to-use front end to the Web, and that revolutionized people’s perceptions. The Internet was being transformed into exactly the network that we had been trying to convince the cable companies they ought to be building. All the stuff we had wanted to do, in generalities, fit perfectly with the way **applications were written, delivered, and used on the Internet**. It was just an incredible accident. And it was patently obvious that the Internet and Java were a match made in heaven. So that’s what we did.”

James Gosling

James Gosling

James Gosling was born near Calgary in 1955. He received his BSc in computer science from the University of Calgary and his PhD from Carnegie Mellon University. In 2007, he was made an officer of the Order of Canada. He is best known as “the father of the Java programming language.”



source: Peter Campbell

Trivial problem

Write a Java program that prints the Java's age.

Step 1

Solve the problem.

Step 2

Write the program.¹

For this you need an editor:

- jEdit
- eclipse
- NetBeans
- Notepad
- ...

¹Instead of program, we often call it an app(lication).

Name of a Java application

According to the Java Language, Specification, the name of an application should be a sequence of letters and digits and the symbols `_` and `$`, starting with a letter.

Code conventions

Rules how to write your code such as

- naming conventions for applications, variables, etc,
- indentation,
- etc.

Appendix C of the textbook contains the code conventions to which you and I will (try) adhere during this course. (We will not adhere to the rule for placement of braces.)

Another example of code conventions can be found at the URL babelfish.arc.nasa.gov/trac/jpf/wiki/devel/coding_conventions.

Code convention for application names

- Capitalize first letter.
- If the name is made up of more than one word, then capitalize the first letter of each.
- If the name is an acronym, then capitalize all letters.

Question

Which of the following class names adhere to the code convention?

- URL
- Sequence_Of_Characters
- Url
- CharacterSequence

```
public class AgeOfJava {  
    public static void main(String[] args) {  
  
    }  
}
```

```
public class AgeOfJava {  
    public static void main(String[] args) {  
  
    }  
}
```

```
public class AgeOfJava {  
    public static void main(String[] args) {  
  
    }  
}
```



```
public class AgeOfJava {  
    public static void main(String[] args) {  
        ...  
    }  
}
```

```
public class AgeOfJava {  
    public static void main(String[] args) {  
  
    }  
}
```

```
public class AgeOfJava {  
    public static void main(String[] args) {  
  
    }  
}
```

```
public class AgeOfJava {  
    public static void main(String[] args) {  
  
    }  
}
```

Do not use magic numbers in expressions.

Magic numbers: numbers different from 0, 1, -1, 2, -2

Why should we not use magic numbers in expressions?

- Obscures the intent of the numbers
 $((100 * 60 * 60) / 9.58) / 1609.34$
- Increases opportunities for subtle errors
- Makes it more difficult for the code to be adapted and extended in the future

Code convention for variables names

- Use lowercase letters.
- If the name is made up of more than one word, capitalize the first letter of each subsequent word.

Question

Which of the following variable names adhere to the code convention?

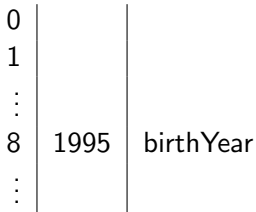
- myVariable
- MyVariable
- my_variable

Memory model

```
int birthYear = 1995;
```

0			
1			
:			
8		00000000	birthYear
9		00000000	
10		00000111	
11		11001011	
:			

```
int birthYear = 1995;
```

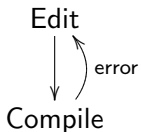



```
System.out.println(age);
```

- System is (the name of) a **class**
- out is (the name of) a **field**
- println is (the name of) a **method**

Step 3

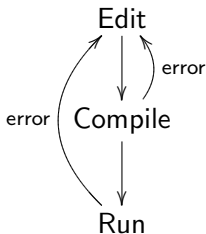
Compile the app: `javac AgeOfJava.java`



eclipse automatically compiles the app.

Step 4

Run the app: `java AgeOfJava`



In eclipse, you press the button with the green circle and the white arrow.

The four steps

- 1 Solve the problem.
- 2 Write the app (using an editor such as jEdit or eclipse).
- 3 Compile the app (done automatically in eclipse).
- 4 Run the app.

How to represent and manipulate data?

Data types: definition

A **data type** consists of

- a **name**,
- a set of **values**, and
- a set of **operations**.

Data types: example

- **name:** `int`
- **values:** `0, 1, -1, 2, -2, ...`
- **operations:** `+, -, *, /, ...`

How many values of type `int` are there?

Question: How many bytes are used to represent a value of type `int`?

Answer:

How many values of type `int` are there?

Question: How many bytes are used to represent a value of type `int`?

Answer: 4

How many values of type int are there?

Question: How many bytes are used to represent a value of type int?

Answer: 4

Question: A byte consists of how many bits?

Answer:

How many values of type int are there?

Question: How many bytes are used to represent a value of type int?

Answer: 4

Question: A byte consists of how many bits?

Answer: 8

How many values of type `int` are there?

Question: How many bytes are used to represent a value of type `int`?

Answer: 4

Question: A byte consists of how many bits?

Answer: 8

Question: How many bits are used to represent a value of type `int`?

Answer:

How many values of type `int` are there?

Question: How many bytes are used to represent a value of type `int`?

Answer: 4

Question: A byte consists of how many bits?

Answer: 8

Question: How many bits are used to represent a value of type `int`?

Answer: $4 \times 8 = 32$

How many values of type int are there?

Question: How many bytes are used to represent a value of type int?

Answer: 4

Question: A byte consists of how many bits?

Answer: 8

Question: How many bits are used to represent a value of type int?

Answer: $4 \times 8 = 32$

Question: How many different values has a bit?

Answer:

How many values of type int are there?

Question: How many bytes are used to represent a value of type int?

Answer: 4

Question: A byte consists of how many bits?

Answer: 8

Question: How many bits are used to represent a value of type int?

Answer: $4 \times 8 = 32$

Question: How many different values has a bit?

Answer: 2

How many values of type `int` are there?

Question: How many bytes are used to represent a value of type `int`?

Answer: 4

Question: A byte consists of how many bits?

Answer: 8

Question: How many bits are used to represent a value of type `int`?

Answer: $4 \times 8 = 32$

Question: How many different values has a bit?

Answer: 2

Question: How many values of type `int` are there?

Answer:

How many values of type `int` are there?

Question: How many bytes are used to represent a value of type `int`?

Answer: 4

Question: A byte consists of how many bits?

Answer: 8

Question: How many bits are used to represent a value of type `int`?

Answer: $4 \times 8 = 32$

Question: How many different values has a bit?

Answer: 2

Question: How many values of type `int` are there?

Answer: 2^{32}

Data types: example

- **name:** int
- **values:** $[-2147483648, 2147483647]$
- **operations:** +, -, *, /, ...

Note that $2147483648 = 2^{31}$.

The operations are typed. For example,

$$\cdot - \cdot : (\text{int} \times \text{int}) \rightarrow \text{int}$$

specifies that the operation $-$ takes two values of type `int` and returns a value of type `int`.

What happens when you subtract one from -2147483648 ?

- **name:** int
- **values:** $[-2147483648, 2147483647]$
- **operations:**
 - $+ : (\text{int} \times \text{int}) \rightarrow \text{int}$
 - $- : (\text{int} \times \text{int}) \rightarrow \text{int}$
 - $* : (\text{int} \times \text{int}) \rightarrow \text{int}$
 - $/ : (\text{int} \times \text{int}) \rightarrow \text{int}$
 - ...

Slightly less trivial problem

Write an app that prints the age of Java as a real number.